Efficient Key Generation for Multicast Groups Based on Secret Sharing

Gajjala Buchi Babu[1], Mutyala Venu Gopal[2], Vellala Sai Srinivas[3], V. Krishna Pratap[4]

1. Computer Science and Engineering, JNTU Kakinada, Akula Sree Ramulu Engineering College, Tanuku, West Godavari Dt, Andhra Pradesh. India Contact no: 9866424858.

2. Science & Humanities, JNTU Kakinada, Sri Sarathi Institute Of Engineering And Technology, Nuzvid, Krishna Dt, Andhra Pradesh. India Contact no: 9949176595.

3. Computer Science and Engineering, JNTU Kakinada, Sri Sarathi Institute Of Engineering And Technology, Nuzvid, Krishna Dt, Andhra Pradesh. India Contact no: 9959956995.

4. Computer Science and Engineering, JNTU Kakinada, Chaitanya Engineering College, Visakhapatnam,
i. Andhra Pradesh. India Contact no: 9985678943.

Abstract:

Secure multicast represents the core component of many web and multimedia applications such as pay-TV, telecon-ferencing, real-time distribution of stock market price and etc. The main challenges for secure multicast is scalability, efficiency and authenticity. In this project, we propose a scalable, efficient, authenticated group key agreement scheme for large and dynamic multicast systems. The proposed key agreement scheme is identity-based which uses the bilinear map over the elliptic curves. Compared with the previously published schemes, our scheme provides group member authenticity without imposing extra mechanism. Furthermore, we give a scalability solution based on the subgroups, which has advantages over the existing schemes. Security analysis shows that our scheme satisfies both forward secrecy and backward secrecy.

***Keywords:* multicast, bilinear pairing, key agreement**

I. INTRODUCTION

Many types of group applications, such as pay per view distribution of digital media, teleconferencing, software updates and real-time delivery of stock market information can benefit from IP multicast which greatly reduced the server overhead and bandwidth usage by enabling source to send a single copy of message to multiple recipients.

One of the main challenges for secure multicast is access control for making sure that only legitimate members of multicast group have access to the group communication. In the passed two or three decades, cryptography has become the well-established means to solve the security problems in networking. However, there are still a lot of difficulties for directly deploying cryptography algorithms into multicasting environment as what has been done for unicasting environment. The

commonly used technique to secure multicast communication is to maintain a group key that is known to all users in the multicast group, but is unknown to any one outside the group. Efficiently managing the group key is a difficult problem for large dynamic groups. Each time a member is added to or evicted from the communication group, the group key must be refreshed. The members in the group must be able

to compute the new group key efficiently, at the same time forward and backward secrecy must be guaranteed. Because the group rekeying is very consumptive and frequently performed

due to the nature of multicast communication, the way to update it in a scalable and secure fashion is required.

II. EXISTING SCHEME

There are several schemes proposed for secure multicast. In this section, we will briefly review some of these schemes.

1. *IOLUS Approach:*

Iolus approach proposed the notion of hierarchy subgroup for scalable and secure mulitcast. In this method, a large communication group is divided into smaller subgroups. Each subgroup is treated almost like a separate multicast group and is managed by a trusted group security intermediary (GSI). GSI connect between the subgroups and share the subgroup key with each of their subgroup members. GSIs act as message relays and key translators between the subgroups by receiving the multicast messages from one subgroup, decrypting them and then remulticasing them to the next subgroup after encrypting them by the subgroup key of the next sub-group. The GSIs are also grouped in a top-level group that is managed by a group security controller (GSC), see Figure 1.

Although Iolus has improved the scalability of the system, because the member join or leave only affect their subgroup only while the other subgroup will not be affected. It has the drawback of affecting data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup, and thereby one key, to another. This becomes even more problematic when it takes into account that the GSI has to manage the subgroup and perform the translation needed. The GSI may thus become the bottleneck.

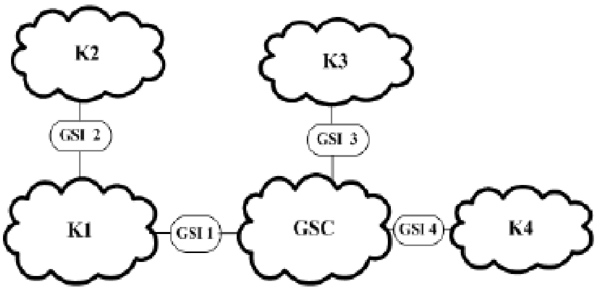

Figure 1: Framework of Iulos

2. *Logical key hierarchy(LKH):*

The logical key hierarchy(LKH) is an efficient approach that supports dynamic group membership. The ideas in this method are identical to convert the cost of communication from linearly to logarithm with the group size of n. In this approach, the group controller (GC) maintains a logical key tree where each node represents a key encryption key (KEK). The root of the key tree is the group key used for encrypting data in group communications and it is shared by all users. The leave node of the key tree is associated with a user in the communication group.

Each user secretly maintains the keys related to the nodes in the path from its leaf node to the root. We call the set of keys that a member knows the key path. Figure 2 shows a sample of key tree. When a member leaves the group, all the keys that the member knows, including the group key and its key path, need to be refreshed. When a member joins the group, GC authenticates the member and assigns it to a leaf node of the key tree. The GC will send the new member all the keys from his/her corresponding leaf node to the root. The main reason for using such a key tree is to efficiently update the group key if a member joins or leaves the group. An optimization of the logical key hierarchy approach is one-way function tree (OFT) proposed by McGrew and Sherman. Their scheme reduces the size of rekeying message from 2 log2 n to log2 n.

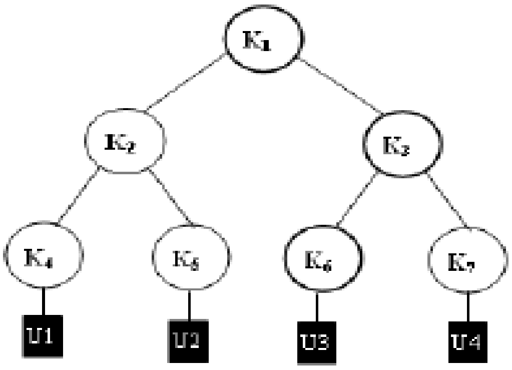

Figure 2: Sample of hierarchical key tree

3. *One-way function chain tree (OFCT):*

This algorithm is known as the one-way function chain tree (OFCT) and it is applied only on user removal. One of the main drawback of LKH and its variants is that they are centralized. In this kind of systems, there is only one entity (GC) controlling the whole group. With the growth of the group member, GC should pay out heavy cost to manage and maintain a huge key tree. This problem is exacerbated when the group has a highly dynamic membership change. Moreover, if the GC aborts, the whole communication group will be affected.

The rekeying method used in has considered a different distribution of keys in the key tree. In this approach, dynamic key hierarchy is used instead of a fixed hierarchy of keys. Also, Boolean function minimization technique to minimize the cost of communication is used. Although the size of rekeying messages and the storage of GC are reduced, their scheme suffers from collusion attack.

III. PROPOSED SCHEME

In this paper, we propose a scalable, efficient, authenticated group key agreement scheme for multicast. Our scheme makes use of the bilinear pairings over the elliptic curves. Our scheme has advantages over the exiting schemes proposed for secure multicast. First, compared with the previously published tree-based schemes our scheme achieves group member authentication without imposing extra mechanism. Since we use an identity tree instead of key tree in our scheme each node in the identity tree is associated with an identity. The leaf node's identity is corresponding to the user's identity and the intermediate node's identity is generated by its children's identity. Hence, in an identity tree, an intermediate node represents a set of users in the sub tree rooted at this node. Next, our scheme solves the scalability problem in multicast systems. Since we divide the large communication group into several smaller subgroups. Each subgroup is independently maintained by the subgroup controller (SGC). In our scheme, even though a subgroup controller fails, it does not affect its subgroup. Because every user in the subgroup can act as the subgroup group controller. This is an amazing feature especially for the groups that has a highly dynamic membership change in mobile and ad hoc networks. Third, in our scheme, the keys used in each subgroup can be generated by a group of key generation centers (KGCs) in parallel. All the members in the same sub group can compute the same subgroup key though the keys for them are generated by different KGCs. This is a desirable feature especially for the large-scale network systems, because it minimizes the the problem of concentrating the workload on a single entity.

1. Security Requirements for Multicast:

We consider dynamic groups where users can join or leave the multicast group at any time. The main security properties of multicast are:

1) Group Key Secrecy guarantees that it is computationally infeasible for a passive adversary to discover any group key.

2) Backward Secrecy is used to prevent a new member from decoding messages exchanged before it joined the group. This property guarantees that a passive adversary who knows a subset of group keys cannot discover the previous group keys.

3) Forward Secrecy is used to prevent a leaving user or expelled group member to continue accessing the group communication. This property guarantees that a passive adversary who knows a subset of old group keys cannot discover the subsequent group keys.

2. Proposed Scheme Operation:

We use the following notation throughout of the remainder this paper shown as Table 1. From our earlier discussion, it can be seen that, in a centralized multicast system, there is only one entity controlling the whole communication group. The group does not rely on any auxiliary entity to perform key generation, key distribution and group rekeying. If there is any problem with the group controller, all the group members in the communication group will be affected. So the group controller is the single point of failure. Additionally, a multicast communication group may has a large number of users, controlled by only one single entity may raise the problem of scalability.

Table 1: Notations

n	number of subgroup members
U_i	i-th group member, $i \in \{1, 2, \cdots, n\}$
ID_i	U_i's identity
N_v^l	l-th level v-th node in an identity tree
Q_j^i	hash value of N_j^i's identity
h	height of an identity tree
P_j^i	private key of the node N_j^i
K_j^i	key generation key of the node N_j^i
BK_j^i	the blinded key of the node N_j^i
KGC_i	i-th key generation center
s_v	the local master key of v-th key generation center

Our protocol directly addresses the problem of reducing the overload of the group controller. We divide the multicast communication group into regional subgroups. Each subgroup is independently managed by a subgroup controller (SGC) like a separate multicast group with its own subgroup key. Thus, when a member joins or leaves the communication group, it joins or leaves only its local subgroup. As a result, only the local subgroup communication key needs to be refreshed and the scalability problem is greatly mitigated. We use a 'group' of key generation centers (KGCs) to share the overall key generation and distribution workload.

In our scheme the task of SGC is just to update the identity tree when there is a membership change in the subgroup and send it to KGCs. Note that this task can be done by any user in the subgroup. All the keys including the users' private keys, blinded keys in our multicast system are generated by the KGCs. Moreover the key distribution is also fulfilled by the KGCs. Using the subgroup key, KGCs can encrypt the message for the subgroup. Although the group members' private keys/blinded keys are generated by distinct KGCs, all members in the same subgroup can generate the same subgroup communication key. This is a significant feature especially for the large and dynamic communication groups. Figure 3 shows the architecture of our mulitcast system.

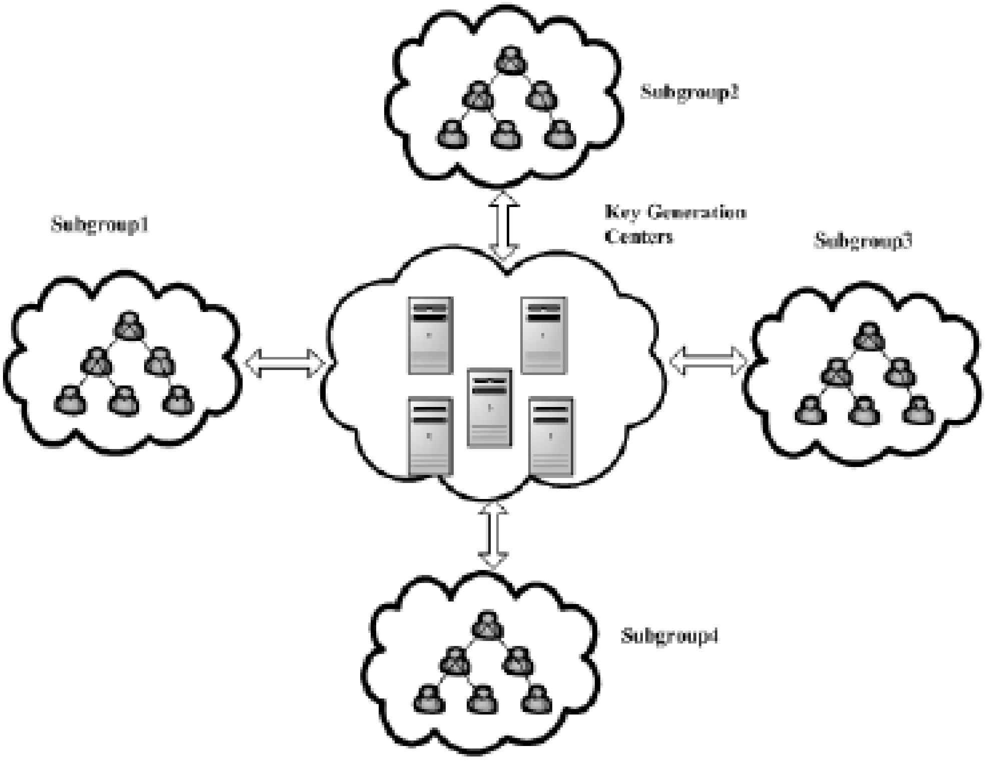

Figure 3: Architecture of our multicast system

The basic idea of our scheme is the usage of an identity tree, where each node in the tree has an identity. The leaf node's identity is corresponding to a user's identity and the interior node's identity is generated from it's children's identity. Figure 4 shows an example of identity tree. A node in the identity tree is also associate with a key generation key (KGK) which is used for generating a parent key. The root node's KGK is used as the group key.

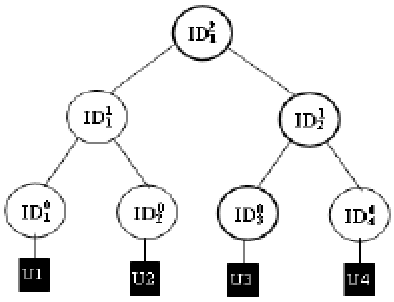

Figure 4: An example of identity tree

3. Member Join Event:

We assume that the subgroup has n users, {U1,U2, · · · ,Un}, when a subgroup receives a joining request from a new user Uj , the SGC searches the nearest leaf node Nx from the root to keep the height of the identity tree as low as possible. Now SGC generates a new node, the member associated with Nx and the new member become the new node's left child and right child respectively. SGC rearranges the levels of the affected nodes and keys in the updated tree T^0 and gives T^0 to KGCs. Then KGCs re compute the private keys of the affected nodes in T0 for the subgroup. Figure 5 shows an example of new user U4 joining the group. A new node $N^1{}_2$ is generated by SGC and becomes the parent of leaves $N^0{}_3$ and $N^0{}_4$

4. Member Leave Event:

When a user Uj leaves the group, all the private keys and KGKs held by nodes in the path from its parent node to the root are compromised and should be updated. This process is handled similarly to the member join event. The only difference is that KGCs compute fewer keys. SGC updates the identity tree by deleting the leaf node corresponding to Uj and rearranges the levels of affected nodes in the updated tree T0. Then SGC sends the updated identity tree T0 to KGCs. KGCs perform the key generation for T0 as described above. For example, see Figure 6, U3 and U4 are deleted from the group. Note that after U3 and U4 leave the group, all the private keys known by U3 and U4 are changed. So U3 and U4 cannot compute the group key in the future. This means that our scheme satisfies forward secrecy.

Furthermore, in the rekeying process, SGC just needs to update the identity tree and send it to KGCs. This task can be done byevery user in the subgroup. So in our multicast system,even if the SGC aborts or leaves, the subgroup will not be affected. This is a significant feature especially for the mobile and ad hoc networks where have a highly dynamic membership change.

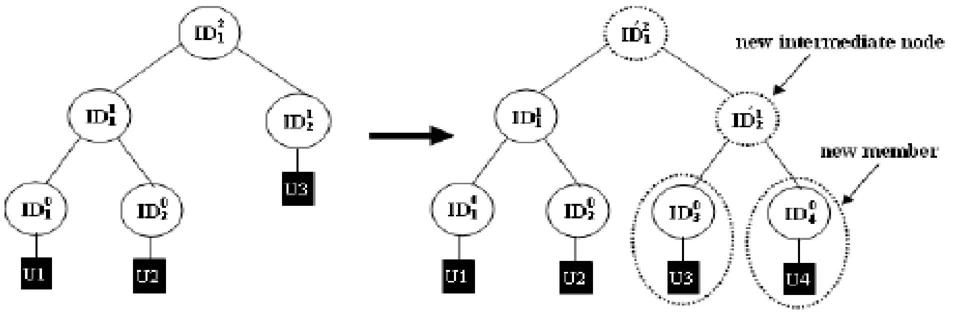

Figure 5: U4 is added to the group

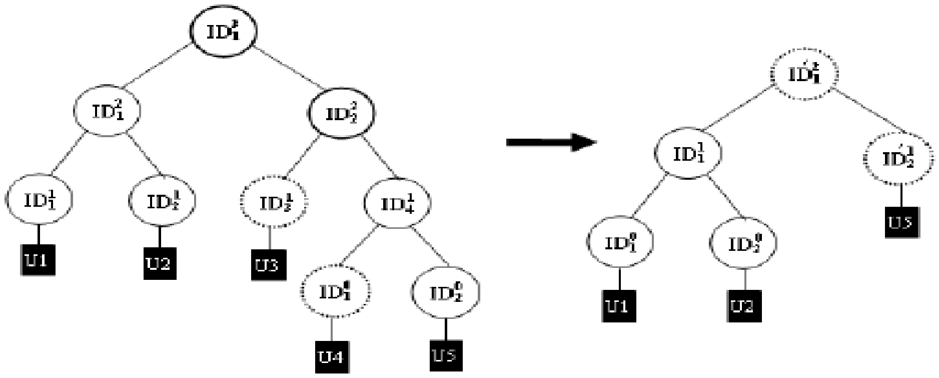

Figure 6: U3 and U4 are deleted from the group

IV. COMPARISON

In this section, we compared our scheme with Iolus and LKH approach and its variants as introduced in Section1.In Iolus, Scalability is achieved by splitting the large group into small groups. If a subgroup controller is failed, only its subgroup is affected. In our scheme, however, even the subgroup controller is failed, its subgroup will not be affected. Because any user in the subgroup can perform the functionality of the subgroup controller. This is a very desirable feature especially for the mobile and ad hoc networks. Furthermore, each subgroup key in Iolus is generated by each group security intermediary (GSI). When the GSIis aborted, how to update the whole system is a difficult problem.

In our scheme all the keys used in the subgroup are generated by a group of KGCs in parallel. Even some of them do not work, it does not have any effect at all. Because the key generation/distribution task can be fulfilled by the remainder KGCs. Compared with the LKH method and its variants, our protocol provides explicitly group member authentication. Since we use the identity tree to achieve this property. In LKH method and its variants, the group controller has a heavy burden to carry out access control policy and maintain a huge key tree. Further, the group controller also has responsibilities to generate, distribute keys used in the group communication. As a result, with the growth of the communication group, the group controller becomes the single bottle neck of the system. When the group controller is not working, the whole communication group becomes vulnerable because the keys, which are the base of the group privacy, are not being generated and distributed. In our scheme, all of these problems are avoided by using a group of KGCs to fulfill these tasks. In our scheme, however, the node in the identity tree is associated with three keys: private key, and key generation key. This makes the user storage in our scheme is larger than that in LKH approach and its variants. However the key generation key on the interior node can be computed by the user, so the user does not need to store it locally. Moreover, the blinded keys as well as the identity tree are the public information; KGCs may store them in a shared storage medium where the users can access to. Using this approach, we achieve the same user storage as in LKH method and its variants.

V. CONCLUSION

We have proposed an efficient, authenticated, scalable key agreement for large and dynamic multicast systems, which is based on the bilinear map. Compared with the previously published schemes in literature, we use an identity tree to achieve the authentication of the group member. Further, our scheme solve the scalability problem in multicast communications. Since a large group is divided into many small groups. Each subgroup is treated almost like a separate multicast group with its own subgroup key. All the keys used in each subgroup can be generated by a group of KGCs in parallel. The intuitively surprising aspect of this scheme is that, even the subgroup controller aborts, it does not affect the users in this subgroup. Because every user in the subgroup can act as a subgroup controller. This is a significant feature especially for the mobile and ad hoc networks

VI. REFERENCES

[1] A. Perrig, D. Song and J. D. Tygar, "ELK, a new protocol for efficient large group key distribution,"IEEE Symposium on Security and Privacy, pp. 247–262, 2001.
[2] M. Scott and P. S. L. M. Barreto, "Compressed pairings," in CRYPTO 2004, LNCS 3152, pp. 140–156,2004.
[3] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," IEEE Transations on Software Engineering, vol. 29, no. 5, pp. 444–458, 2003.
[4] M. Steiner, G. Tsudik, and M. Waidner, "Cliques: a new approach to group key agreement," IEEE Conference on Distributed Computing Systems (ICDCS'98), pp. 380–387, 1998.
[5] D. Wallner, E. Harder, and R. Agee, Key Management for Multicast: Issues and Architectures, RFC 2627, Internet Engineering Task Force, June 1999.
[6] C. K. Wong and S. Lam, "Secure group communications using key graphs," in SIGCOMM '98, pp. 68–79, 1998.